Identifying key signals from data using Anomaly Detection

Archish Ramesh Babu[1], Agastya Teja[2], Gopal Seshadri[3]

Abstract

Supervised Machine Learning techniques for Anomaly Detection require previously tagged data. Manually tagging the points as Anomaly or Not an Anomaly is time consuming and prone to manual error. We are building multiple unsupervised Anomaly detection models that can perform anomaly detection in any time series data without the need of manual tagging the data. We have selected our algorithms in such a way that they could be scaled to multidimensional data with minimal modifications.To measure the performance of our unsupervised models we have identified the anomalous points from the Yahoo Time Series data set and compared them against the actual labels. We have also used the bench marked the performance of our algorithms using the credit card data available on Kaggle. In addition to all the models we have also implemented our framework on frequency of twitter count for a topic in specific time period to identify any anomalous behaviour. If any anomaly was identified we have tried to explain the driver for those anomalies in the form of bag of words. .

Keywords

Unsupervised Anomaly Detection — Time Series — Twitter Data

[1] *Computer Science, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

Introduction

The world is becoming more interconnected each day due to the disruption caused by social media, IoT, globalization to name a few things. Companies are opening up new branches, products are being introduced daily producing metrics and new health devices are in use, this has led to new data being generated primarily time series data. Since data is getting generated in real time, in order to stay competitive, companies need to analyze the data quickly and take appropriate action. Anomaly detection is a popular concept which has been widely used with supervised learning. But the problem with this approach or any supervised method is that it takes time for individuals to tag anomalies and is also subjected to manual errors. To be able to make quick decisions and make the work of tagging easier, we propose to build an unsupervised model which could be used to detect potential anomalies. This process does not require any labelled data and takes much lesser time than a human sitting through loads of data to identify anomalous cases for every new dataset. We will be achieving this with the help of multiple preexisting unsupervised anomaly detection packages namely

1. Isolation Forest
2. One-class SVM
3. K-means
4. Max Voting Method
5. Twitter Time series Anomaly Detection Module

We have primarily focused on comparing the performance of these models on time series data and comparing their results with the Anomaly detection package from twitter. The time Anomaly detection time series data released by Yahoo is a good source to benchmark various algorithms. The data is based on traffic from yahoo and the anomalies are manually tagged by employees. In addition to comparing the performance of these anomaly detection models on time series data we have also compared their performance of credit card data available on Kaggle.

1. Background

Anomalies are deviations from expected behaviour. Anomaly detection is a technique where we use models to identify data that is not in line with the rest of the data. Anomalies could be point Anomalies which are produced by abrupt spikes or dips. Moreover, Anomalies could also be a general change in trend or small spikes. Anomaly detection has been applied by multiple industries for versatile tasks to improve their performance. Banks use it for Fraud Detection, identifying transactions made by stolen cards and alerting their customers and the police. Product based companies have begun to track the performance of their spare parts using Sensors and are using Anomaly Detection to identify if all the parts are working well and if replacement of any part is needed. Early replacement of parts could stop a lot of wastage that would be created if the product breaks down. Anomaly detection is also being used in the medical files for identifying diseases like Breast cancer. New fields have also started using Anomaly detection for innovative applications.

Anomaly detection has been primarily run on supervised learning methods. Supervised Algorithms are a class of Machine Learning models which learn from preexisting data containing labels, so to run Anomaly Detection with supervised Learning we need to feed the model sufficient data of Anomalies along with non-Anomalous data for it to learn. Post which we have to tune the parameters and repeatedly update the model so that it is able to identify any new type of anomalies. Many researchers have tried finding anomalies using supervised machine learning models, in the paper titled "Anomaly Detection with Machine learning" written by Hanna Blomquist and Johanna Möller it was demonstrated that AdaBoost and Support Vector Machines can be used for Anomaly Detection for text classification tasks. It could also be noted that since the data which they used was highly unbalanced they evaluated their model based on Precision and Recall. This entire process is cumbersome and time-consuming plus companies would not be able to perform anomaly detection till all these steps have been completed. This has worked well in the past but we will need a much faster and scalable framework for anomaly detection due to pace at which data is being generated.

This is where unsupervised Machine learning methods shine. Unsupervised Machine Learning models do need any labelled data to train upon and can run as soon as the data is being generated. A lot of unsupervised Machine Learning Algorithms have come up in recent times namely One class Support Vector Machine, Isolation Forest, K Means. In the paper "Isolation based Anomaly Detection" Fey Tony Liu, Kai Ming ting and Zhi-Hua Zhou have extensively studied isolation forest methods for detecting anomalies. Their study shows that Isolation Forest is scalable to big data with high dimensionality and it is much faster than other algorithms for finding anomalies.

Companies like Twitter have built an unsupervised anomaly detection module for time series data. The problem is that there is not much research has been done on using these algorithms for time series data. We are building a general anomaly detection framework that would work for both the time series data and multidimensional data as well. We have build a general purpose anomaly detection framework that would work for any time series data and could also be extended to multidimensional data as well. This would reduce the complexity of choosing the algorithms according to the data and people would be quickly able to churn out results.

Once the Anomalies have been identified the task does not stop there as we will need to understand the root cause of the Anomaly and fix it and that would involve gathering more data about the points before and after the anomaly.

2. EDA

The time series data was explored by looking at multiple plots of the data, its distribution and based on slicing it by days, hours, weeks etc.

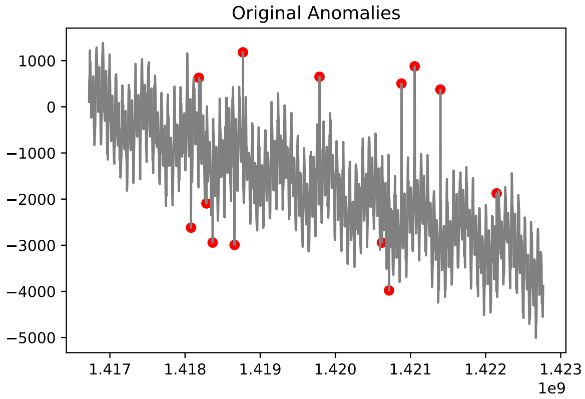

Figure 1. Timeseries data with tagged Anomalies provided by Yahoo

Based on our analysis we have observed some of the data had trends and seasonality. We have created separate variable in our data set to deal with all these factors. Below is how we dealt with trend and seasonality.

2.1 Trend

Time series data has a lot of trend, the sales of a new product is expected to increase over time. Now the increase should not be tagged as an anomaly but the sudden spike or dip in the sales should be correctly identified as soon as possible for the business teams to take necessary action. Majority of the yahoo benchmark data sets have trend in them. Below is an image of time series data before and after trend removal.

The trend was removed using a simple linear Regression model, we modelled the value of the time series data against the position of the time series data ranging from 1 to n where

1 was the first value sorted by time and n was the last value. Hence the dependent value was the value and the independent value was the created row number. The coefficient was the trend value. We them multiply the trend value with the row number and subtract it against the value.

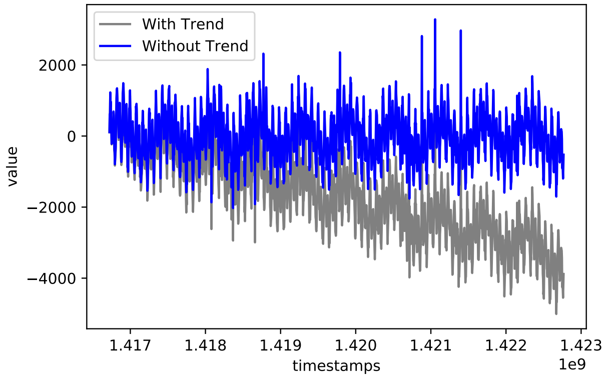

Figure 2. Data before and after trend removal

2.2 Seasonality

In addition to trend the time series data also has a lot of seasonality.For example, the traffic during the weekends is generally higher on websites due to the additional time or it could be vice verse in other cases. Also, there is a lot of difference in sales during various hours of the day. The sales of a product between 11 pm to 3 am will be much lower than the sales of the product between 4 pm to 6 pm. All these factors need to be considered which dealing with Anomalies. Below is an image of how the traffic varies by the day of a week.

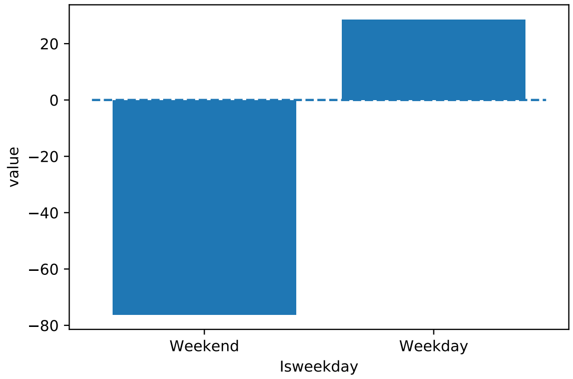

Figure 3. Variation in value by weekday and weekend

The effect of Seasonality has been dealt with by adding a variable for each of the seasonal parameter like days, hours in our data. We could also incorporate yearly event like Christmas as an another variable if necessary.

3. Baseline Model

The most intuitive way to detect anomalous points for univariate data like time series data is to use its statistical properties such as mean and standard deviation to find out extreme points in the data. In this method we implement this by finding the average value of the entire data and find out points which fall outside certain standard deviation(for the purpose of this exercises we are using 3 standard deviations) . These points represent extreme points and can roughly e approximated as anomalous points and this will be our base model.

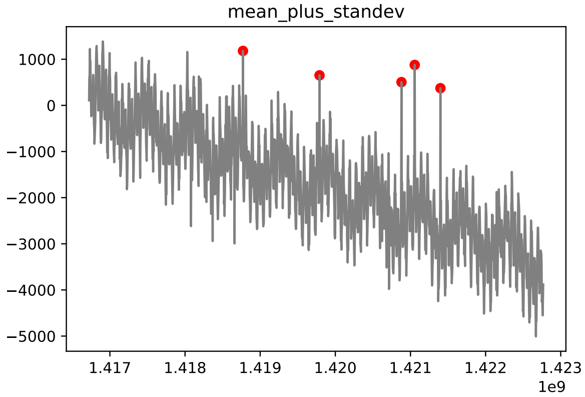

Figure 4. The baseline model is not able to identify majority of the Anomalies

The problem with this approach is that it fails to consider the effect of seasonality, trend and is also influenced a lot by the existing outlier points. Hence it was not able to identify even one outlier.

4. Models and Methodology

The problem with time series methods like ARIMA is that it not very scalable to multidimensional. It becomes time-consuming to identify the p - autoregressor, d - differences for stationarity, q - number of lagged forecast errors values for ARIMA models and also it cannot scale to multidimensional data. We need to need a more automated method like Isolation Forest, SVM, K Means to get results quicker and could be scaled to multidimensional data as well.

4.1 Model Evaluation

Anomalous points are very rare and hence the data for anomaly detection is highly unbalanced. Because of this property,the accuracy metric do not work well. The best method to compare Anomaly Detection methods would be use the Recall and F1 Score. The Recall metric measures the number of Anomalous points that the model is able to correctly tag to the total number of anomalous points in the data, we would like this value to be closer to one. The other metric is the F1 score, it is the harmonic mean of the recall and precision. The precision is the ratio of the number of correctly tagged anomalous points to the total number of points tagged as anomaly, we would also like this number to be closer to 1, because it would mean we have less number of False Positive.

4.2 Rolling mean with standard deviation

The main drawback of the Mean + Standard Deviation model is that the mean is heavily influenced by fluctuations and hence is not good representative of the data. To avoid this a rolling window is chosen and average is taken over those points in the window. This rolling window precedes the point which we are trying to detect anomaly or not. So, when we traverse through the dataset from the beginning to the end trying to predict moving average we will not sufficient points in the rolling window for the beginning points. To avoid this boundary cases we convolute the original dataset with an array of weight 1/(length of window) which will give us moving average of the window . The output of this convolution will result in window – 1 extra points and these points are removed.

In the same way rolling standard deviation is calculated by summing up squares of differences between data points and moving average over the points in the rolling window and divided by number of points. The immediate point after the rolling window is then compared with rolling mean by checking its deviation and the points are tagged as anomalous based on amount of deviation.

4.3 Isolation Forest

Isolation forest is a relatively new technique and is built on the concept of the Decision Trees and work on the facts that anomalous points need much less information to identify them. The model tries to find anomalous points by randomly splitting each variable by looking the depth of the tree, the points identified with less depth are considered as anomalies since they would be away from the major chuck of the data. This is intuitive since the anomalous points are far away from the actual data cluster and we could reach them with just a few nodes. The isolation forest also runs with linear complexity and is performs parallel processing. One issue with Isolation forest is that the model needs to be trained with some anomalous data and only then it would be able to correctly tag similar data points during the scoring process.

4.4 One Class SVM

One-class Support Vector Machine is a variant of the regular SVM model. In SVM method we can identify multiple classes of data but in one class SVM we can only use it to separate two classes, One class belonging to the normal data and the other class belonging to the anomalous data. The model is trained on data with no anomalies. Moreover, the model is not affected even if it is trained with a few anomalous points because the model has an inbuilt parameter that creates a soft boundary which minimizes the effect of a few data points on the model prediction. This method is very effective but the problem is with the runtime. We have run the One-class SVM model with the default parameters for this exercise.

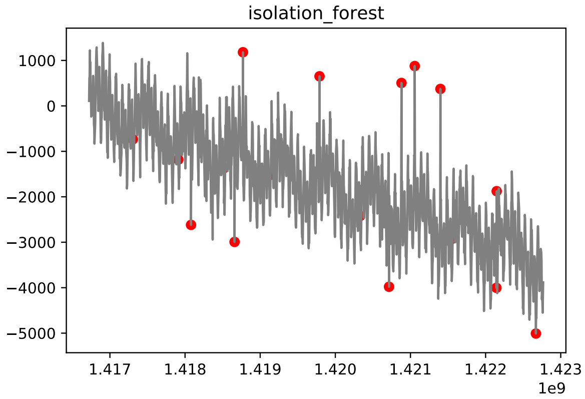

Figure 5. IsolationForest is able to tag majority of the Anomalies

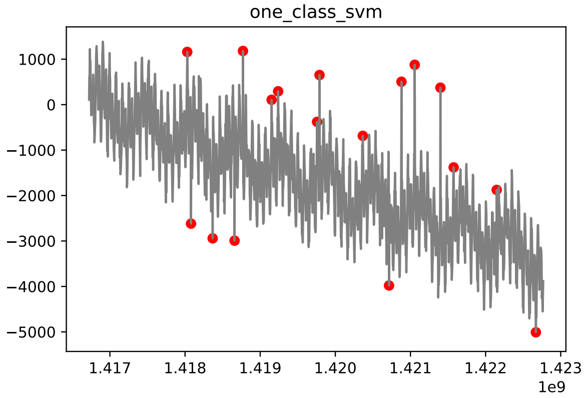

Figure 6. One class SVM is able to tag majority of the Anomalies

4.5 K Means

This is a distance-based anomaly detection method, in this method we first specify the number of clusters to be created. Once these clusters are formed based on the distance between the point and the cluster the points which have a very large distance from the cluster are tagged as anomalous points. We have set the cluster size to 4 for all the data, the value was chosen since the data had very fewer dimensions.

4.6 Max Voting

Each of the classifier algorithms have their own advantages and disadvantages and would able to identify different anomalies.In order to boost the true positive and minimize the false positive, we have created a voting algorithm where all the above algorithms vote on a points to be an anomaly or not. If a point is tagged as an anomaly by 2 or more algorithms among the SVM, Isolation Forest and K means. We have shortlisted these three algorithms because they are the best performing ones.

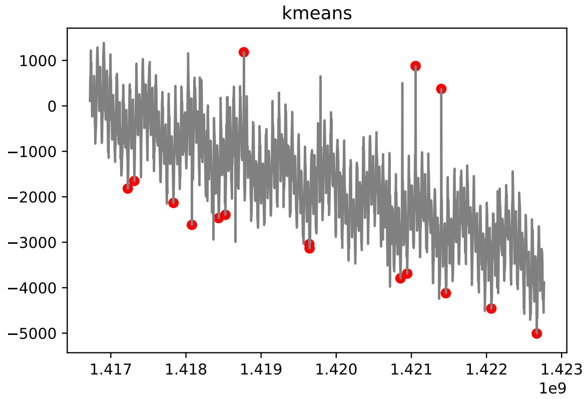

Figure 7. K Means Anomoly Detection algorithm has a lot of False Positives

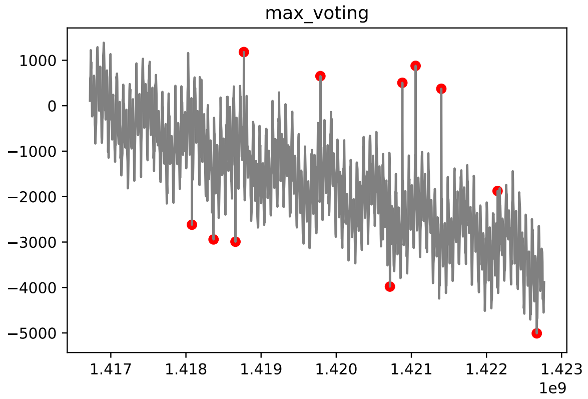

Figure 8. MaxVoting is able to tag majority of the Anomalies correctly with less False Positives

4.7 Twitter package

This package was developed by Twitter and made opensource on the R programming environment. This package uses the Seasonal Hybrid ESD method for detecting anomalies. The back end of this method is a combination of time series decomposition and a Generalized EST test. This model failed when the data had trend but when the trend was removed this method was very effective and was able to accurately identify majority of the anomalies on all the data set with very low False Positive. This module is only available in the R programming environment currently.

5. Experiments and Results

All the algorithms have been run on 67 different Time Series Anomaly Detection dataset provided by Yahoo and the predicted anomalous points have been compared with the actual anomalous points for benchmarking. The average results are available on table 1. Based on the results in table 1 (from the experiment done on 67 different time series data), we can observe that One-class SVM performs better the other methods for time series data. The One-class SVM method outperforms the moving average plus standard deviation by 4 times in terms of recall and also has a higher F1 Score. The Max voting is close to the SVM in terms of Recall Metric but it has a much higher F1 score, this is because the other algorithms reduce the number of False Positives from the data.

Table 1. Average Recall and F1 score for 67 time series data

Method	Recall	F1 Score
Twitters AD package	1.000	0.999
One-Class SVM	0.845	0.576
Max Voting(not including twitter package)	0.771	0.665
Isolation Forest	0.609	0.417
Moving Avg Rolling standev	0.599	0.601
K-means	0.393	0.270
Mean plus standev (Baseline)	0.373	0.509
Moving Avg plus standev	0.208	0.296

Isolation Forest is performing a bit poorly than expected because the data size is very small and also it has very few features in them. This is probably because taking a sample from an already small population might lead to unexpected behaviour. The performance of the K Means algorithm is a lot lower than expected because the number of clusters to be used for each of the data varies a lot and it extremely hard to tune it without labelled data.

We can also observe that Twitters Time series algorithm performs extremely well on the Yahoo dataset. The only disadvantage of this algorithm is that it is currently not available in Python and can only be used with R programming language. Now that we have identified the anomalous points based on our algorithms and validated the results using the Yahoo dataset as a benchmark, we can move to an application in the real life. Social Media is gaining a lot of traction these days and becoming the primary medium through which people express their views is through twitter. We have used the Tweepy package in Python to scrape Twitter data after creating a Twitter Developer account. We can run our Anomaly Detection Algorithm's on popular topics to check if there is any change in their trends. This is an extremely useful tool to have for many different purposes. For Example, Pharma companies can see if a lot of customers are talking about any side effects, by quickly identifying these trends we can save a lot of lives. Product companies can track the sentiment about products and take corrective action. Design teams can look for trending ideas on Twitter and launch products based on those ideas, For example, if the topic ”lemon” was trending as a popular item the company could launch a lemon flavoured product.

For the purpose of a demo, we have considered the chain Target to check if there is any change to its trend, upon running the count vs time data for target through our model have identified a few anomalous points. The twitter package is also listing the same points as anomalous, hence the model is scaling well for twitter count data as well.

Looking at only the points as anomaly or not is enough,

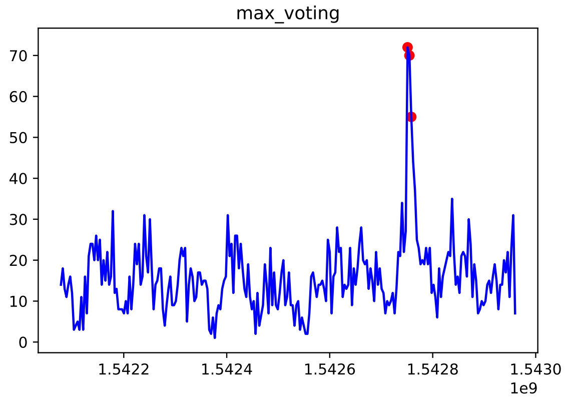

Figure 9. MaxVoting Algorithm run on real twitter data

Table 2. Recall and AUC score for Kaggle's Credit card Fraud Detection data

Method	Recall	AUC
One-class SVM	0.876	0.954
Isolation Forest	0.829	0.945
K-means	0.843	0.897

to further understand the reason for the spike we would need to look at the word cloud. Based on the word cloud for the anomalous point we can see that the reason for the spike is Black Friday sale. This framework could be extended to other trending topics and we could create alters based on these anomalies.

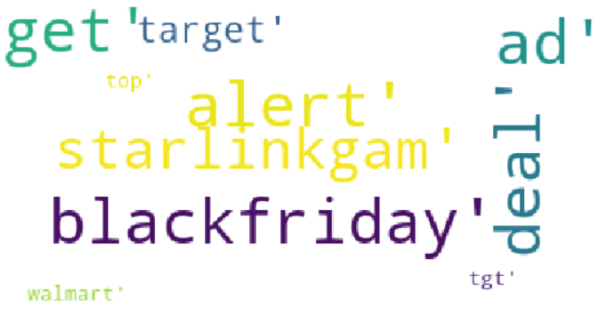

Figure 10. Wide Picture

Now that we have analyzed one-dimensional data we can move on to multidimensional dataset. To test the performance of our data on multidimensional data, we have use the Credit Card Fraud Detection dataset from Kaggle and compared the performance of the well-performing algorithms

Based on the above table we can see that the models scale well for multidimensional data as well. The F1 score for these algorithms was in the lower end due to high False Positives and hence we have used the AUC score which is a good substitute.

6. Summary and Conclusions

Unsupervised Learning for Anomaly Detection is a need of the day due to vast of amount new data being generated from various sources.To implement anomaly detection using standard Supervised Machine Learning techniques will take a lot of time and effort to set up. Unsupervised Machine Learning provides an opportunity to quickly set up a framework and analyze new data without much or any effort at all. The anomalies identified by Unsupervised Machine Learning models could be later used to setup a supervised Machine Learning model.

Many Unsupervised Anomaly Detection Machine Learning algorithms have come up recently and it is necessary to set up a Standard Unsupervised Anomaly Detection Framework that could run on all data both univariate(time series) and multivariate data. We have compared the performance of some popular Algorithms like One Class SVM, Isolation Forest, Kmeans and Twitters AD package on Yahoos standard AD time series data and bench marked their performance against simple mean plus standard deviation. All of these algorithms significantly outperform the benchmark based on Recall. One Class SVM is the best performing model followed by the Isolation Forest Model and K Means. The One class SVM is able to identify around 80 percent of the Anomalies correctly with a very low false positive. The Max voting algorithm which takes the output of these models to identify has the highest F1 score due to the increase in precision caused by the reduction in False Positive Score.

All these models also perform quite well on Multidimensional as well. The above-mentioned algorithms had a Recall above 80 percent and an AUC score over 0.9. For multidimensional data also the One-class SVM performs slightly better than the isolation forest but the run time of the Isolation Forest is much lower than the One-class SVM. Based on these observations we can conclude that One-class SVM and Isolation Forest models work well with both time series and multidimensional data but there is room for improvement. One algorithm that has come up recently is the Autoencoder and needs to be explored further. Autoencoder is an unsupervised neural network algorithm which takes a data and compresses it and then reconstructs the same data. So if we train our Autoencoder with sufficient non anomalous data it will be able to reconstruct the input with minimal change and now if anomalous data passes through the trained Autoencoder the reconstructing will be poor which will result in poor accuracy between the original data and the reconstructed. Hence we can identify anomalous points based on the error between the original and reconstructed data.

Acknowledgments

We are very grateful to Professor Gregory J.E. Rawlins and the Associate Instructors for imparting valuable Machine Learning Concepts throughout the semester. The concepts such as Random Forests, Support Vector Machines, K Means were the foundations to algorithms such as Isolation Forest, one class SVM and K Means which have been implemented in our project. A big thank you to Manoj Joshi, our project mentor for guiding us throughout the project and for encouraging us to extend the Anomaly detection algorithms for multidimensional data credit card data available on Kaggle.

We are also grateful to Christopher Torres-Lugo for helping us with identifying the correct algorithms for this project based on his previous experience in the field of Anomaly Detection. A special thank you to Zeeshan Ali Sayyed for evaluating our idea motivating us to proceed with our project in its very nascent phases. Finally, we would like to thank Indiana University Bloomington for giving us all the resources for the successful completion of this project.

References